UNIVERSIDADE da MADEIRA

CTeSP CURSOS TÉCNICOS SUPERIORES PROFISSIONAIS

Tecnologias e Programação de Sistemas de Informação

Introdução à linguagem de programação Java

Arquitetura de Dispositivos | David Jardim

Cofinanciado por:

MADEIRA 14-20

REGIÃO AUTÓNOMA DA MADEIRA

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Social Europeu

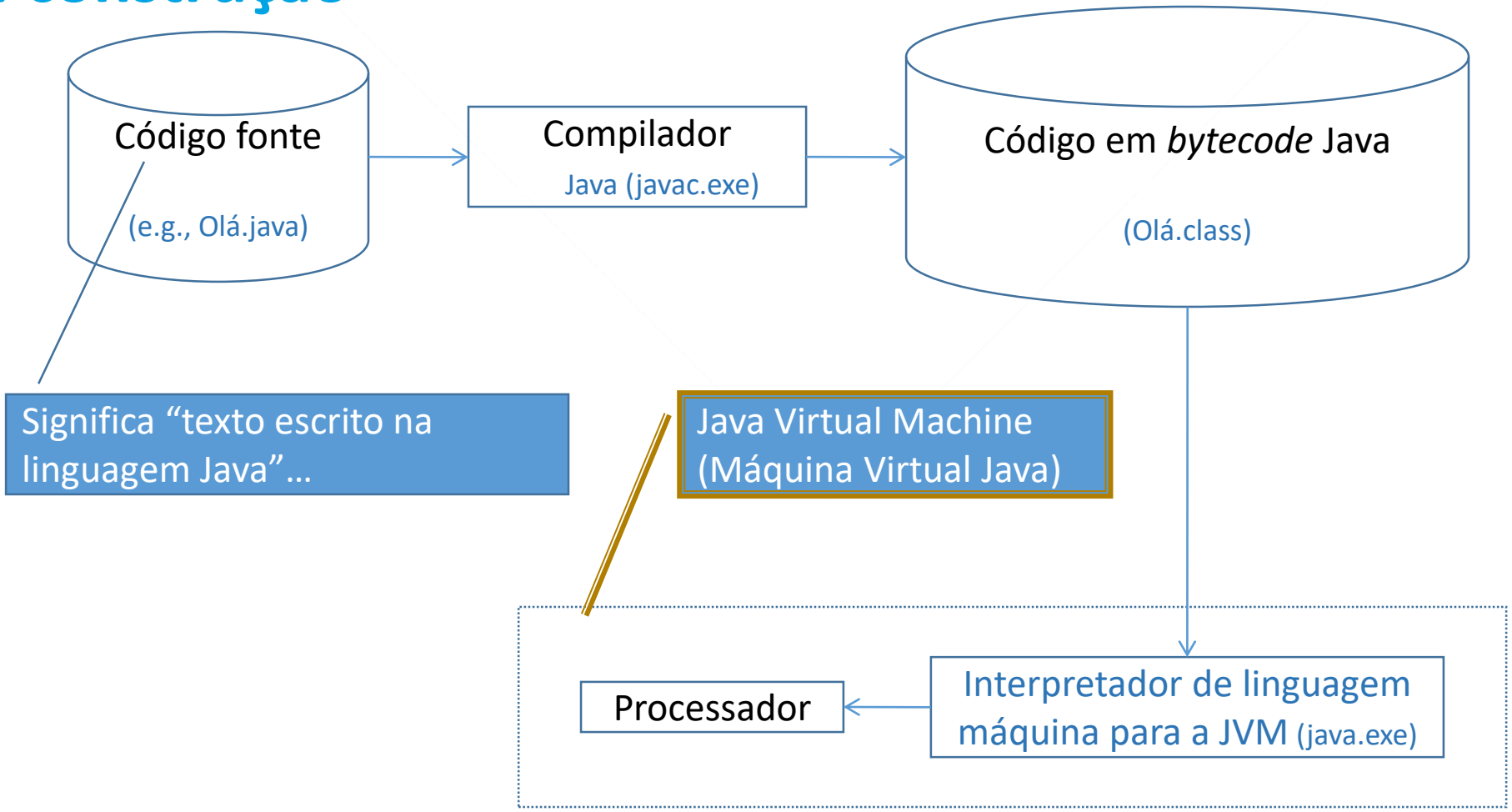
Fases da resolução de problemas

- Especificar problema [humano]
- Desenvolver algoritmo que o resolva [humano]
- Concretizar algoritmo na linguagem de programação [humano]
- Traduzir programa para linguagem máquina [compilador, numa máquina]
- Executar programa para caso particular [máquina física ou virtual]

Java

- Linguagem de programação orientada para objectos
  - Paradigma dominante
  - Existem outros paradigmas
    - Programação procedimental
    - Programação funcional
    - Programação lógica
- Independente da plataforma: programas executáveis em diferentes sistemas operativos
- Muito utilizada na indústria

Java: construção



Variáveis

- Declaração e inicialização obrigatórias!
  - Declaração indica nome e tipo (conjunto dos valores)
  - Inicialização indica valor inicial
- Utilizadas para guardar valores
- Atribuição altera valor de variável

Variáveis: boas práticas

- Declarar tão próximo quanto possível da primeira utilização
- Declarar de modo a minimizar âmbito da variável
- Inicializar logo que possível
- Não inicializar apenas “para calar o compilador”

Variáveis

Pseudocódigo	Java	Observações
-	`int  i = 1;`	Declaração com inicialização.

-	`int  gcd;`
	
	`if (m < n)`
	`    gcd = m;`
	`else`
	`    gcd = n;`
		Declaração sem inicialização seguida de duas inicializações alternativas.

primo ← 2.	`prime = 2;`	Atribuição (se a estiver já inicializada).
i ← *i* + 1.	`i = i + 1;`	Incrementação (há melhores formas…).
n ← *temporária*.	`n = temporary;`	

Variáveis: tipos

Tipo	Descrição	*Bits*	Exemplos
`int`	Subconjunto dos inteiros.	32	`int numberOfStudents;`
`double`	Vírgula flutuante. Subconjunto dos racionais.	64	`double averageGrade;`
`boolean`	Booleanos ou lógicos.	-	`boolean isPrime;`
`char`	Caracteres.	16	`char response;`
`String`	Texto, cadeias de caracteres.	-	`String studentName;`

Variáveis: outros tipos

Tipo	Descrição	*Bits*
`byte`	Pequeno subconjunto dos inteiros.	8
`short`	Subconjunto dos inteiros, entre `byte` e `int`.	16
`long`	Subconjunto dos inteiros, maior que `int`.	64
`float`	Vírgula flutuante. Subconjunto dos racionais , menor gama e menor precisão que `double`.	32

Variáveis e identificadores

- Nomes de variáveis são identificadores

- Identificadores não podem ser repetidos no mesmo contexto

Identificadores: formato

- Constituídos por
  - letras (a, À, ε, …)
  - dígitos (0 a 9)
  - _
  - $
- Primeiro caractere não pode ser dígito
- Maiúsculas e minúsculas são distinguidas
- Não podem ser palavras-chave do Java (e.g., `for`, `while`, `int`, `if` ou `for`)

Identificadores: convenções para variáveis

- Primeira palavra em minúsculas
- Restantes palavras com maiúscula inicial
- Exemplo: `numberOfStudents`

Identificadores: boas práticas

- Em língua natural
- Sem abreviaturas
- Gramática correcta
- Adequados à entidade que identificam
- Claros
- Significativos
- Expressivos

Operadores

Pseudocódigo	Java	Significado	Tipo
∧	&&	e	Booleano
∨	||	ou	Booleano
¬	!	não	Booleano
<	<	menor	Relacional
≤	<=	menor ou igual	Relacional
>	>	maior	Relacional
≥	>=	maior ou igual	Relacional
=	==	igual	Comparação
≠	!=	diferente	Comparação

Instrução de seleção

Pode-se omitir as chavetas quando contêm apenas uma instrução.

Pseudocódigo	Java

Se *m* < *n*, então
  mdc ← *m*,
senão,
  mdc ← *n*.
	`if (m < n) {`
	`  gcd = m;`
	`} else {`
	`  gcd = n;`
	`}`

Se *nota* < 10, então
  …
senão,
  se *nota* < 12, então
  …
  senão,
  …
	`if (grade < 10) {`
	`  …`
	`} else if (grade < 12) {`
	`  …`
	`} else {`
	`  …`
	`}`

Instruções de iteração

Pseudocódigo	Java

inicialização
Enquanto *guarda*, fazer
  acção
  progresso.
	initialization
	`while (`*guard*`) {`
	  action
	  progress
	`}`

inicialização
Fazer
  acção
  progresso
enquanto *guarda*.
	initialization
	`do {`
	  action
	  progress
	`} while (`*guard*`);`

Instruções de escrita no ecrã

Pseudocódigo	Java
Escrever *algo* no ecrã.	`System.out.print(`*something*`);`
Escrever *algo* no ecrã e mudar de linha.	`System.out.println(`*something*`);`

Funções

Pseudocódigo	Java

Função mínimoDe(*m*, *n*)
Se *m* < *n*, então
Devolve *m*.
senão,
Devolve *n*.
	`static int minimumOf(final int m, final int n) {`
	`  if (m < n)`
	`  return m;`
	`  else`
	`  return n;`
	`}`

x ← mínimoDe(5, 7).	`int x = minimumOf(5, 7);`

Rotinas

- Podem ser funções ou procedimentos
- Funções – calculam e devolvem algum valor
- Procedimentos – realizam uma dada acção

A ver mais tarde…

Métodos

- Em programação orientada para objectos as rotinas são conhecidas por métodos

- Podem ser de classe ou de instância

- Métodos de classe – com `static`

- Métodos de instância – sem `static`, a ver mais tarde

Hello world!

```
public class Greeter {

    public static void main(final String[] arguments) {
        System.out.println("Hello world!");
    }
}
```

- Ficheiro com o código fonte de classe tem de ter mesmo nome que classe e extensão `.java`

- Método principal `main(...)` é primeiro a ser invocado ao se executar um programa

Comentários

- Usados para clarificar código menos claro

- Java ignora
  - texto entre // e o fim da linha e
  - texto entre /* e */

- "Comentários" iniciados com /** são documentação, que veremos mais tarde

Operadores de atribuição especiais

Atribuição simples	Atribuição especial	Observação
`sum = sum + item;`	`sum += item;`	Atribuições especiais semelhantes: • `-=` • `*=` • `/=` • `%=`
`i = i + 1;`	`++i;` `i++;`	Versões prefixo e sufixo equivalentes *neste contexto*. Não são equivalentes *em geral*!
`j = j - 1;`	`--j;` `j--;`	

Vetores (arrays)

- Sequências de itens com comprimento fixado durante construção

- Declaração
  - *tipoDosItens*[] *nome*;

- Declaração, construção e inicialização
  - *tipoDosItens*[] *nome* = new *tipoDosItens*[*comprimento*];

construção

- Exemplos
  - `final double[] grades = new double[numberOfStudents];`
  - `final int[] sizes = new int[numberOfClasses];`
  - `final int numberOfStudents = grades.length;`

Vetores (arrays)

- Itens identificados por índices
- Primeiro item: índice 0 (zero)
- Último item: índice `vector.length - 1`
- Exemplos
  - `int firstSize = sizes[0];`
  - `int lastSize = sizes[sizes.length - 1];`

Vetores

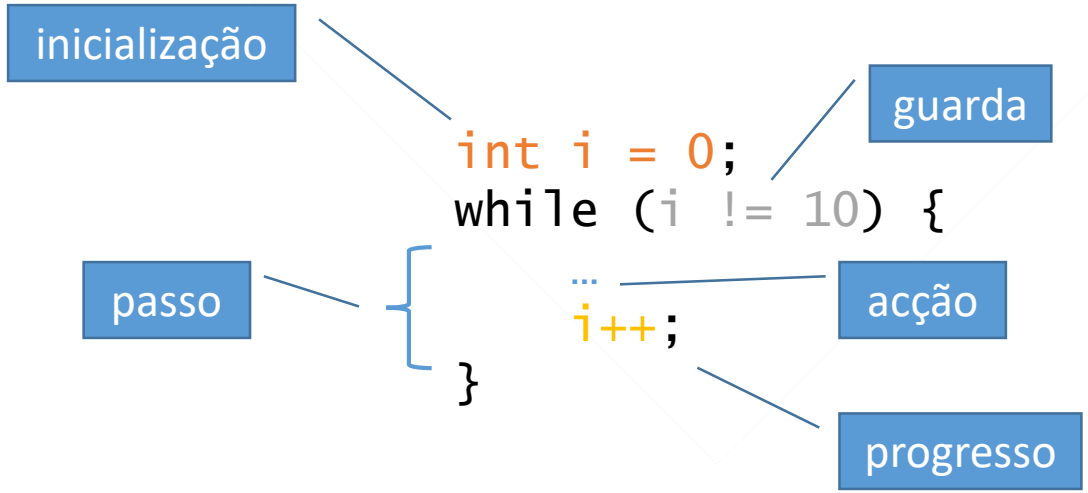
Pseudocódigo	Java
valores ← lista(5, 0).	`int[] values = new int[5];`
tamanhoDe(*valores*)	`values.length`
valor ← *valores*[2].	`int value = values[2];`
valores[0] ← 7.	`values[0] = 7;`

Construção de vetores

Valores por omissão (zero) e atribuições	Valores explícitos

`final int[] values = new int[3];`
`v[0] = 0;`
`v[1] = 2;`
`v[2] = 4;`
	`final int[] values = {0, 2, 4};`

Ciclos: While



Ciclos: for

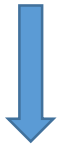
- Alternativa a `while`
- Tipicamente usado com iteradores
- Útil para manipular de vectores

```
int i = 0;
while (i != 10) {
    …
    i++;
}
```

```
for (int i = 0; i != 10; i++) {
    …
}
```

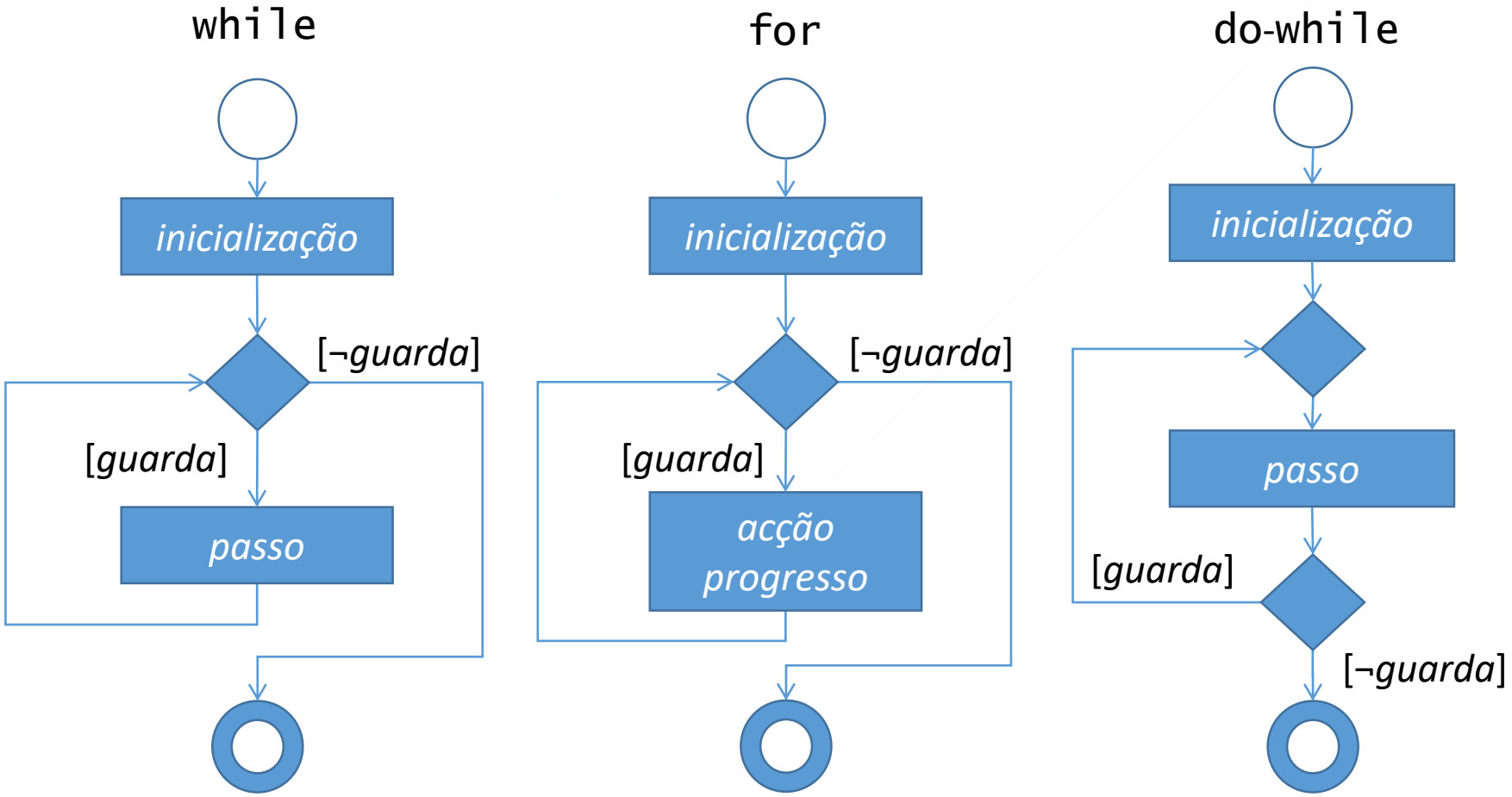
Ciclos: while vs. for

```
double sum = 0.0;
int i = 0;
while (i != grades.length) {
    sum += grades[i];
    i++;
}
```



```
double sum = 0.0;
for (int i = 0; i != grades.length; i++) {
    sum += grades[i];
}
```

Ciclos: lado-a-lado…



Blocos: se instrução única, dispensáveis

Possível	Preferível

`while (values[i] != value) {`
`    i++;`
`}`
	`while (values[i] != value)`
	`    i++;`

`if (hour == 0) {`
`    hour = 23;`
`} else {`
`    hour--;`
`}`
	`if (hour == 0)`
	`    hour = 23;`
	`else`
	`    hour--;`

Operador de seleção

Seleção	Operador Ternário

`if (m < n)`
`    maximum = n;`
`else`
`    maximum = m;`
	`maximum = m < n ? n : m;`

instrução

operação

Exemplo de método função

```
/**
 * Returns maximum of the items in array.
 *
 * @param array  array whose maximum will be returned.
 * @returns          the maximum of the items in array.
 * @pre array must have at least one item
 */
public static int maximumOf(final int[] array) {
    int maximum = array[0];

    for (int i = 1; i != array.length; i++)
        if(maximum < array[i])
            maximum = array[i];

    return maximum;
}
```

A reter

- Java
  - Variáveis
  - Instrução de selecção `if-else`
  - Instruções de iteração `while` e `do-while`
  - Rotinas vs. funções e procedimentos
  - Rotinas e métodos
  - Métodos de classe (`static`) vs. métodos de instância
  - Escrita no ecrã com `System.out.println(…)`
  - Método principal `main(…)`
  - Comentários e documentação
  - Vetores
  - Ciclos


UNIVERSIDADE da MADEIRA
CTeSP
CURSOS TÉCNICOS
SUPERIORES PROFISSIONAIS